MMSound.Dll

Version 1.7

Vector Informatik GmbH, Ingersheimer Straße 24, 70499 Stuttgart

Tel. (0711) 80670-0, Fax (0711) 80670 555,

Email can@vector-informatik.de, Internet http://www.vector-informatik.de

Subsidiaries

France

Vector France SAS

168, Boulevard Camélinat
F-92240 Malakoff

Tel.: +33 1 4231 4000
Fax: +33 1 4231 4009

http://www.vector-france.com

Japan

Vector Japan Co., Ltd.

Nishikawa Bldg. 2F
3-3-9 Nihonbashi, Chuo-ku
J-103-0027 Tokyo

Tel.: +81 3 3516 7850
Fax: +81 3 3516 7855

http://www.vector-japan.co.jp

Sweden

VecScan AB

Fabriksgatan 7
S-41250 Göteborg

Tel.: +46 031 79901 35
Fax: +46 031 79903 05

http://www.vecscan.com/

USA

Vector CANtech, Inc.

Suite 550
39500 Orchard Hill Place
USA-Novi, Mi 48375

Tel.: +1 248 449 9290
Fax: +1 248 449 9704

http://www.vector-cantech.com

Addresses of our distributors can be found on our website:

http://www.vector-informatik.com

Content

1 Overview

MMSound.DLL is a CAPL DLL that provides basic audio functionality with some additional audio file management.

In short, the following features are available:

- Playback and transport of audio files of ".wav" and ".mp3" format
- Recording of audio files
- Management of multiple audio files using handles, f.e. to implement play list functionality
- Reading out title, artist and genre information out of the id3 tag of ".mp3" files

Chapter 3 describes the installation and integration of the DLL in a CANoe or CANalyzer application.

Chapter 4 contains a detailed description of all available functions.

2 License

3 Installation

The MMSound.Dll extends the CAPL function set. Therefore the CAPL compiler must know about the location of the Dll. One of the following options must be applied.

1.1 CAPL Directive

Copy the MMSound.dll to your configuration folder and add a `#pragma library` directive in section `includes` of the CAPL program. The file path of the Dll relative to the CAPL file has to be quoted.

```
includes
{
  #pragma library ("MMSound.dll")
}
```

Note: This option is available for CANoe/CANalyzer versions equal or newer than version 6.1 only.

1.2 Application Extension

Make the MMSound.dll available for all CAPL programs by assigning the Dll to the CANoe/CANalyzer application in general.
Therefore it is recommended to copy the Dll file into the Exec32 folder of the CANoe/CANalyzer installation, although it's also possible to access it from anywhere else, e. g. the folder of your configuration.
Then open CAPL DLL configuration dialog of CANoe/CANalyzer via **Configuration | Options… | Extensions | CAPL DLL**. A list of all CAPL Dlls assigned to the application is displayed. Click on **[Add…]** to browse for the Dlls location and select the file and confirm with **[OK]**.

4 Function reference

4.1 mmsndOpen

Syntax	`dword mmsndOpen(char[] filename)`
Description	Opens the given audio file and returns a handle, that can be used to start playback of the file.
Parameters	`filename` The name of the file to be opened. The file must be of ".wav" or ".mp3" format. If a standard path was provided via `mmsndSetMediaFilePath()` previously, any `filename` containing a relative path or the name of the file only, is concatenated to this path for loading. In case `filename` contains an absolute path, the standard path is ignored.
Return values	> 0 : Handle for the audio file, in case the file exists and can be read.
	0 : File does not exist or can't be read.

4.2 mmsndOpenFolder

Syntax	`dword mmsndOpenFolder(char[] filepath, dword outHandleList[], dword handleListLen)`
Description	Searches for all ".wav" and ".mp3" audio files within a given directory and returns a list of handles referencing the audio files found. A handle can then be used to start playback of the corresponding file using `mmsndPlay()`. Note that this function can be called multiple times to add the audio files of several directories to the handle list of the DLL. To delete the current list of handles, `mmsndCloseAll()` can be called at any time. Additionally, the file name for a handle can be retrieved by a call to `mmsndGetFilePath()` with the handle value as input parameter.
Parameters	`filepath` The name of the folder to be searched for audio files of ".wav" or ".mp3" format. If a standard path was provided via `mmsndSetMediaFilePath()` previously, any `filepath` containing a relative path or the name of a folder only, is concatenated to this path for loading. In case `filepath` contains an absolute path, the stan-

	dard path is ignored.
	`outHandleList`
	Array that is filled by the function with handles for all audio files found within the given path.
	`HandleListLen`
	Size of the array passed over to the function as handleList parameter.

Return values	> 0: Number of audio files found in the directory, which corresponds to the number of handles written into `handleList`.
	0 : No files where found or given filepath is invalid

4.3 mmsndPlay

Syntax	`dword mmsndPlay(dword handle)`
	`dword mmsndPlay(dword handle, dword fromPos)`
	`dword mmsndPlay()`
	`dword mmsndPlay(char[] filename)`
	`dword mmsndPlay(char[] filename, dword fromPos)`
Description	Starts playback of an audio file via a handle previously returned by a call to `mmsndOpen(),` or by direct specification of a file name. Additionally, the starting playback position can be provided. The signature without any parameter starts playback of the first handle in the list, if no previous call to `mmsndPlay()` with a handle parameter occurred. If playback of a file referenced by a handle is running or is paused, playback of that file will be restarted. If playback of a file referenced by a handle has finished, playback of the file referenced by the next handle in the list will be started. Generally, any currently running playback will be stopped.

Parameters	`Handle`
	Handle of the audio file to be played.
	`fromPos`
	Playback starting position in milliseconds
	`filename`
	The name of the file to be loaded. The file must be of “.wav” or “.mp3” format. If a standard path was provided via `mmsndSetMediaFilePath()` previously, any `filename` containing a relative path or the name of the file only, is concatenated to this path for loading. In case `filename` contains an absolute path, the stan-

	dard path is ignored.
Return values	1 : Playback started
	0 : Handle or filename is invalid or file can't be opened. When using the `fromPos` parameter, the value provided may be out of range. Set the verbose level to at least 1 to get a error message describing the reason.

4.4 mmsndPause

Syntax	`dword mmsndPause()`
Description	Pauses currently running playback. When recording, the recorded data will be written into the file specified in the `mmsndStartRecord()` call.
Parameters	-
Return values	1 : Playback paused successfully.
	0 : Playback was not active

4.5 mmsndContinue

Syntax	`dword mmsndContinue()`
Description	Continues playback after it was paused previously by a `mmsndPause()` call.
Parameters	-
Return values	1 : Playback continued successfully.
	0 : Pause was not active

4.6 mmsndStop

Syntax	`dword mmsndStop()`
Description	Stops currently running playback or recording. When recording, the recorded data will be written into the file specified in the `mmsndStartRecord()` call.

Parameters	-
Return values	1: Playback or recording stopped successfully.
	0: Playback or recording is already stopped. In case of recording, the DLL may be unable to write the data into the file. Set the verbose level to at least 1 to get a error message describing the reason.

4.7 mmsndClose

Syntax	`dword mmsndClose(dword handle)`
Description	Unloads the audio file referenced by `handle`. If playback of this file is currently running, it is stopped. `handle` gets invalidated.
Parameters	`Handle` Handle of an audio file previously opened via `mmsndOpen()`.
Return values	1 : Audio file successfully unloaded.
	0 : Invalid handle.

4.8 mmsndCloseAll

Syntax	`dword mmsndCloseAll()`
Description	Closes all currently opened audio files. Current playback of an audio file referenced by a handle is stopped. All handles are invalidated.
Parameters	-
Return values	Number of file handles closed.

4.9 mmsndGetCurrHandle

Syntax	`dword mmsndGetCurrHandle()`
Description	Closes all currently opened audio files. Current playback of an audio file referenced by a handle is stopped. All handles are invalidated.
Parameters	-

Return values	The handle of the currently played audio file or 0, if the audio file was specified directly via a path.

4.10 mmsndGetFilePath

Syntax	`dword mmsndGetFilePath(char outFilePath[], dword filePathLen)`
	`dword mmsndGetFilePath(dword handle, char outFilePath[], dword filePathLen)`
Description	Retrieves the file path of the currently played audio file or the file path referred by a given handle.

Parameters	`outFilePath`
	String parameter filled by the function with the result file path.
	`filePathLen`
	Size of the string variable passed over to the function for `filePath.`
	`handle`
	Handle value previously returned by `mmsndOpen()` or `mmsndO-penFolder()`

Return values	1 : Valid path successfully written into `outFilePath`
	0 : No current file played or invalid handle

4.11 mmsndStartRecord

Syntax	`dword mmsndStartRecord(char[] filename)`
Description	Starts recording an audio file from the line-in of the audio hard-ware. Recording can be stopped by a call to `mmsndStop()`.
Parameters	`filename` The name of the file in which the recorded data will be written. The filename must have a “.wav” ending. If a standard path was provided via `mmsndSetMediaFilePath()` previously, any `filename` containing a relative path or the name of the file only, is concatenated to this path for loading. In case `filename` con-tains an absolute path, the standard path is ignored. Note that any existing file with the specified name will be overwrit-ten.

Return values	1 : Recording successfully started.
	0 : Recording couldn’t be started. Set the verbose level to at least 1 to get a error message describing the reason. .

4.12 mmsndSetPlayPos

Syntax	`dword mmsndSetPlayPos(dword pos)`
Description	Sets the current playback position while playback is running or paused.
Parameters	`Pos` New playback position in milliseconds.
Return values	1 : New playback position set
	0 : Playback position couldn’t be set, because player is not in playback or paused state.

4.13 mmsndGetCurrPlayPos

Syntax	`dword mmsndGetCurrPlayPos()`
Description	Returns the current playback position in milliseconds. When recording, the current length of the recording is returned. Note, that due to the buffering of the audio hardware of the PC, the position returned is usually aligned to multiples of the currently set buffer size.
Parameters	-
Return values	Playback position or current length of recording in milliseconds. 0, if playback is stopped.

4.14 mmsndGetTrackLen

Syntax	`dword mmsndGetTrackLen(dword handle)`
Description	Returns the length of an already opened audio file.
Parameters	`handle` Handle of an audio file previously opened via `mmsndOpen()`.
Return values	Length of the audio file in milliseconds.

4.15 mmsndGetFileInfo

Syntax	`dword mmsndGetFileInfo(char filePath[], dword infoField, char outFieldValue[], dword fieldValueLen)`
	`dword mmsndGetFileInfo(dword handle, dword infoField, char outFieldValue[], dword fieldValueLen)`
	`dword mmsndGetFileInfo(dword infoField, char outFieldValue[], dword fieldValueLen)`
Description	These functions try to read the title, artist or genre information out of the id3 tag of a “.mp3” file. It will not work on an “.wav” file, since no such information is stored in this file format. The three signatures differ only in the way the input file is provided: Either a direct file path, a handle of a already opened audio file or the currently played audio file.

Parameters	`filePath`
	
	File path to “.mp3” file. Processing of given path takes media path set previously by mmsndSetMediaPath() into consideration: Relative paths or simple file names are concatenated to that path.
	
	`infoField`
	
	Type of information to be retrieved. Currently, the following values can by provided:
	
	0: Title
	
	1: Artist
	
	2: Genre
	
	`outFieldValue`
	
	String parameter filled by the function with the result file information.
	
	`fieldValueLen`
	
	Size of the string variable passed over to the function for `outFieldValue.`

Return values	1 : Request information is successfully copied into `outFieldValue`
	0 : An invalid path or handle was given as input, or the requested information is not available, because the corresponding frame or field couldn’t be found in the id3 tag of the “.mp3” file,

	or the file is a “.wav” file.

4.16 mmsndGetPlayerState

Syntax	`dword mmsndGetPlayerState()`
Description	Returns the current player state:
Parameters	
Return values	0 : Stop 1 : Playback 2 : Pause 3 : Record

4.17 mmsndSetMediaPath

Syntax	`void mmsndSetMediaPath(char[] path)`
Description	Sets a standard file path that is used by all subsequent calls to `mmsndPlay()` and `mmsndOpen()` as a prefix to the file name or path provided. .
Parameters	`path` File path to a directory containing audio files to be loaded.
Return values	-

4.18 mmsndSetVerbose

Syntax	`dword mmsndSetVerbose(dword verboseLevel)`
Description	Sets the level of information to be displayed in the CA-Noe/CANalyzer Write window.
Parameters	`verboseLevel` == 2: All actions of the Dll are reported. == 1: Only errors are reported == 0: No reporting at all
Return values	The current verbose level

5 Example

```
includes

{

  #pragma library ("MMSound.dll")

}


variables
{
  dword gHandle1;
  dword gHandle2;
  dword gRet;
  char  gFileSound01[256]  =  “Track01.wav”;
  char  gFileSound02[256]  =  “Track02.wav”;
}


on start
{
  mmsndSetMediaPath("C:\\Sound\\");
}


on stopMeasurement
{
  mmsndStop();
  mmsndClose(gHandle1);
  mmsndClose(gHandle2);
}


on key 'o'
{
  gHandle1 = mmsndOpen(gFileSound01);
  gHandle2 = mmsndOpen(gFileSound02);
}


on key 's'
{
  mmsndStop();
}


on key '1'
{
  gRet = mmsndPlay(gHandle1);
}


on key '2'
{
  gRet = mmsndPlay(gHandle2);
}


on key 'c'
{
  gRet = mmsndClose(gHandle1);
}
```